3*d*MPC
Model Predictive Controller
Version 1.1

Product Guide

3BSE 018 935R101

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions.

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaTalk, and 3*d*MPC are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.
MATLAB is a registered trademark of The MathWorks Inc.
Windows and Windows NT are registered trademarks of Microsoft Corporation.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products AB. ABB Automation Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

CE MARKING

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

Copyright © ABB Automation Products AB 2000

3BSE001264/E
Template: 3BSE001286/E

TABLE OF CONTENTS

Chapter 1 - Overview

1.1 Introduction 1-1
1.2 Positioning and Features 1-1
1.3 Product Benefits 1-2

Chapter 2 - Functional Description

2.1 General 2-1
  2.1.1 Model Predictive Control 2-2
  2.1.2 Observer Based Model Predictive Control 2-3
2.2 On-line Functionality 2-3
  2.2.1 Controller 2-3
  2.2.2 Graphical User Interfaces 2-5
2.3 Off-line Engineering Tools 2-7
  2.3.1 Configuration Tool 2-7
  2.3.2 Pre-Processing Tool 2-7
  2.3.3 Modeling Tool 2-8
  2.3.4 Tuning Tool 2-9

Chapter 3 - Software Components

3.1 On-Line Components 3-1
3.2 Off-Line Components 3-1
3.3 Directory Structure 3-1

Chapter 4 - Requirements and Compatibility

4.1 Hardware Requirements 4-1
4.2 Operating System Requirements 4-1
4.3 Software Requirements 4-1
4.4 Compatibility 4-1

Chapter 5 - Environmental Immunities 5-1

Chapter 6 - Technical Data and Performance

6.1 Technical Data 6-1
6.2 Performance 6-1
  6.2.1 Memory Consumption 6-1
  6.2.2 Execution Time 6-1

Chapter 7 - Ordering 7-1

ILLUSTRATIONS

Figure 1-1. Feedback control system. 1-1
Figure 2-1. Overview of the different components of the 3*d*MPC. 2-1
Figure 2-2. One of the operator interface displays. 2-6
Figure 6-1. Estimated memory usage. 6-2
Figure 6-2. Estimated execution time. 6-2

TABLES

Table 3-1. 3*d*MPC sub-directories. 3-1
Table 4-1. Minimum recommended hardware for 3*d*MPC 4-1
Table 6-1. Maximal number of signals 6-1
Table 7-1. 3*d*MPC Licence 7-1
Table 7-2. 3*d*MPC Software Delivery Media 7-1
Table 7-3. Documentation on 3*d*MPC. 7-1
Table 7-4. 3*d*MPC size upgrade 7-2
Table 7-5. 3*d*MPC version upgrade 7-2

Chapter 1 Overview

1.1 Introduction

The product 3*d*MPC is intended for multivariable feedback control and optimization of an industrial process that have many input and output signals. Inputs in the form of manipulated variables affecting actuators, and outputs in the form of process variables from sensors. Model Predictive Control, MPC, is a widely used technique for advanced control. For processes with strong interaction between different signals this technique can offer substantial performance improvement compared with traditional single-input single-output control strategies.

1.2 Positioning and Features

The product 3*d*MPC controls a multivariable processes by a combination of feedback and feed-forward. The control system under consideration is shown in Figure 1-1. The controller determines the manipulated outputs, OUT, based on actual measurements of process variables, PV, and feed forward signals, FF, and of operator defined parameters, and external inputs. The process variables can be assigned set-points that are the target for the feedback control law. The feed forward signals are measurable disturbances acting on the process that can be used for feed forward.

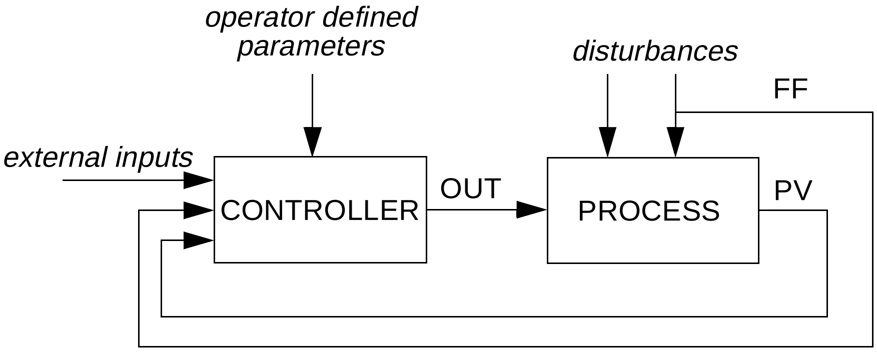


Figure 1-1. Feedback control system.

The number of variables handled by the controller is limited as follows:

- The number of manipulated outputs (OUT) is limited to 16.
- The number of process variables (PV) is limited to 32.
- The number of feed-forward variables (FF) is limited to 16.

These limitations are due to the windows for the graphical user interaction, and not due to algorithm limitations. The 3*d*MPC product is based on the model predictive controller (MPC) concept. This means that, at each sample the manipulated outputs are obtained from constrained optimizations. The loss functions in these problems penalizes a weighted sum of squared control-errors and moves in the manipulated outputs. The constraints in the optimization comes from limits on the manipulated outputs and from limits on the process variables.

The 3*d*MPC product consist of both on-line and off-line components. The on-line components consist of algorithms for control, of a data-logging function, and a function for the generation of excitation signals for system identification. The on-line components also include graphical user interfaces for operator interaction and start and stop of the on-line algorithms.

The 3*d*MPC product is intended to run on a PC under Windows NT 4. It is connected to an OCS/DCS through an OPC server running in the same PC as the on-line controller. The graphical interface can run on the same PC or on a network PC.

The off-line engineering tools are implemented as a MATLAB toolbox and require MATLAB 5.

1.3 Product Benefits

Some benefits of the product are:

- The control algorithm is a three degree of freedom MPC. The three degree of freedom means that the responses to set-point changes, the responses to measurable disturbances (feed-forward variables), and the responses due to the feedback control can be tuned independently.
- The chosen model representation allows handling of integrating or unstable processes without specific precautions.
- The various signals in the controller can operate in independent modes. This means that the controller can be reconfigured on-line to handle various operating conditions, e.g different production, or sensor or actuator failures.
- The controller integrates functions for non-linear transformation on each signal. This is useful for e.g compensation of non-linear sensors and non-linear actuators.
- Macro functionality for coordinated setting of several parameters in the controller.
- There is a flexible handling of constraints for the various signals in the controller. Signal limits (constraints) could be defined by the operator, or they could be external inputs to the controller. Limits for process variables could be expressed either as absolute limits or as limits relative to the set-points.
- Parameter scheduling for handling non-linear or time-varying processes.
- Supervision of process variables and feed forward variables to detect failure situations.
- Definition of critical variables that are required to operate in automatic modes. If such a signal is not part of the optimization procedure it is not meaningful or allowed to operate the controller in any automatic modes at all.
- Modeling tools that allow process identification based on closed loop or open loop experiments.
- A graphical operator interface for on-line supervision and operator interaction.
- An engineering interface with elaborate analysis tools for simple and safe commissioning.
- Internal data-logging functions.

Chapter 2 Functional Description

2.1 General

3*d*MPC consists of on-line components for control and operator interaction, and of off-line engineering components for configuration, modeling, tuning, and analysis.

The on-line components of the 3*d*MPC consist of a controller object and graphical interfaces for user interaction. The on line controller object has an OPC interface for data exchange with an OCS/DCS.

The off-line engineering components are implemented in MATLAB. They are accessed using graphical user interfaces.

The components of 3*d*MPC is shown in Figure 2-1. The figure also shows the interfaces to the OCS/DCS using OPC servers residing in the local NT station.

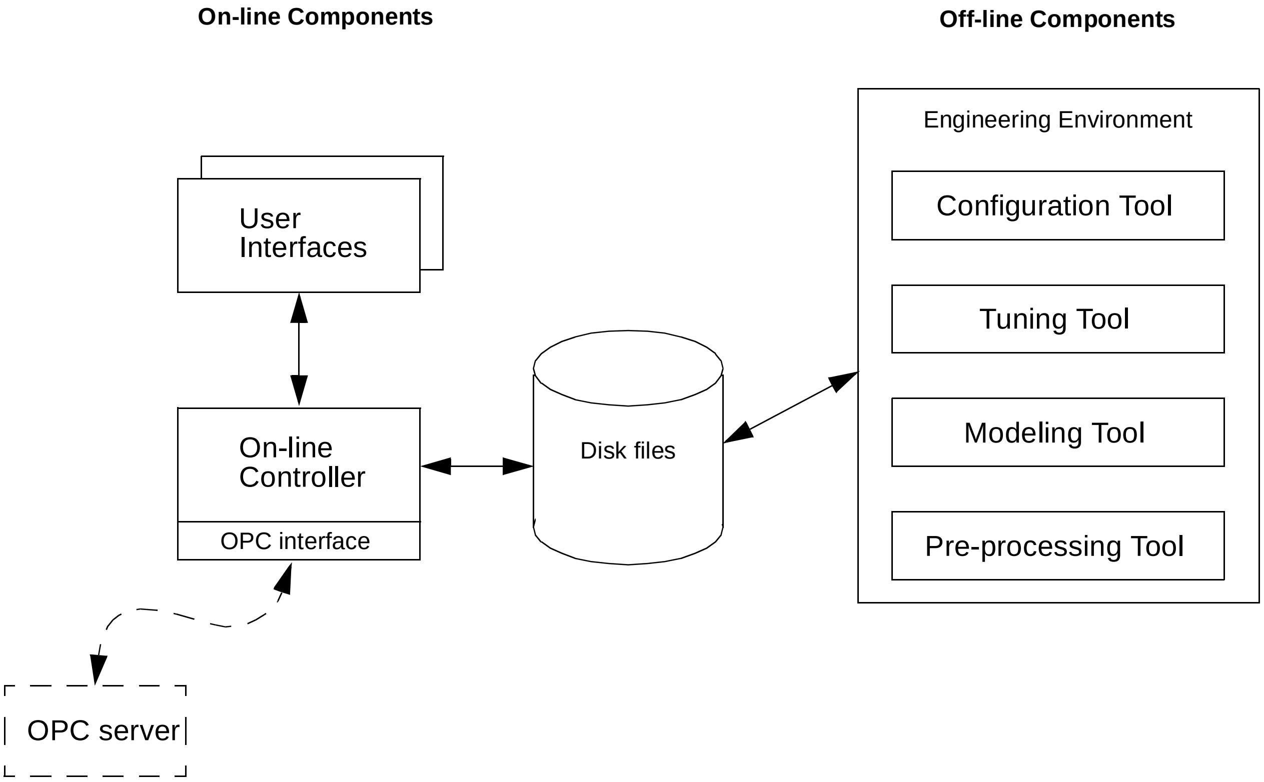


Figure 2-1. Overview of the different components of the 3dMPC.

2.1.1 Model Predictive Control

Model predictive control (MPC) is a fairly general class of algorithms for feedback and feed-forward control. Various MPC techniques have in common that a model of the process, subject for control, is used to predict the effects in process variables due to actual and future changes in manipulated outputs and feed-forward variables.

The dynamic model

$$y(k) = G(q)u(k)$$

that describes the real world process relates how manipulated outputs *u(k)*, from the operator or from a controller, affects the actual and future process variables *y(k)*. Further, *G(q)* is the discrete time transfer function for the process model.

The process variables are measured signals that should follow desired set-points *r(k)*. The model predictive control objective could be defined as follows:

> Find a sequence of current and future increments for manipulated outputs that minimizes a weighted sum of future squared control errors and a weighted sum of increments in the sequence of manipulated outputs, while limits for manipulated outputs and limits for predicted process variables are considered.

This objective can be expressed more precise mathematically: Find a sequence of increments for manipulated outputs $\Delta u(k+i)$ over a control horizon of *m* samples to minimize the objective function

$$J = \sum_{i=1}^{p} \|\Gamma e(k+i)\|_2^2 + \sum_{i=0}^{m-1} \|\Lambda \Delta u(k+i)\|_2^2$$

subject to constraints

$$\begin{aligned} y_{lo} \le y(k+i) \le y_{hi} \qquad & \forall i \in [1, p] \\ u_{lo} \le u(k+i) \le u_{hi} \qquad & \forall i \in [0, m-1] \\ \Delta u_{lo} \le \Delta u(k+i) \le \Delta u_{hi} \qquad & \forall i \in [0, m-1] \end{aligned}$$

In this minimization the control errors *e(k)=r(k)-y(k)* are considered over a prediction horizon of *p* future samples. The constraints for process variables *y(k)* are considered over a prediction horizon of *p* samples, and the constraints for manipulated outputs *u(k)* and their increments are considered over a control horizon of *m* future samples. In this procedure, the dynamic model is used for the prediction of future values for the process variables.

2.1.2 Observer Based Model Predictive Control

In an observer based model predictive controller, like this one, the dynamic model is described as a discrete time state-space model

$$x(k+1) = Ax(k) + B_u u(k) + B_d d(k)$$

$$y(k) = Cx(k)$$

where $x(k)$ is the state vector, $d(k)$ are measurable disturbances used for feed forward. The matrices A, B_u, and B_d define the model. These parameters are determined using the modeling tools provided in the 3*d*MPC product.

2.2 On-line Functionality

The on-line components of the 3*d*MPC consist of a controller object and graphical interfaces for user interaction.

2.2.1 Controller

Operational Modes

The inputs and outputs of the controller can operate in independent modes. This means that some signals in the controller can work in automatic mode while other signals can work in manual mode. The modes for the signals can be changed on-line.

The operational mode for a manipulated output determines if the actual value is determined by the MPC algorithm, or by the operator, or by an external input to the controller.

The operational mode for a process variable determines how the variable is used in the controller and how the corresponding set-point is determined. The set-point could be determined by the operator or by an external input to the controller.

The operational mode for a feed-forward variable determines how the variable is used in the controller.

Critical Variables

Any variable in the controller can be marked as critical. This means that the variable is so important that it must take active part in the MPC algorithm, otherwise it is not meaningful or allowed to use automatic control at all. The controller will then have all manipulated outputs operating in manual mode.

Three Degree of Freedom Dynamic Optimization

The controller can be configured to have different dynamic responses to

- set-point changes
- changes in measurable disturbances
- feedback control.

To achieve this, three optimization problems, of the type described in Section 2.1.1, Model Predictive Control, are solved with appropriate control error formulations. For each of these three control error formulations, the dynamical response is defined by equivalent control error factors for the process variables and by equivalent move factors for the manipulated outputs.

Static Optimization

After the dynamic optimizations has been performed an algorithm is employed to drive the manipulated outputs towards desired values. This algorithm will only have effect in cases where there is ambiguity in how manipulated outputs are chosen for desired targets for the process variables. The static optimization algorithm penalizes weighted differences between the actual and desired values for manipulated outputs.

Constraint Handling

A main feature of model predictive controllers is the ability to handle constraints, not only for manipulated outputs but also for process variables.

Two sets of soft constraints with priorities can be defined for each process variable: One set defines constraints as absolute values, and one set defines constraints as values relative to the set-points. The constraints could be defined either by external inputs or by the operator.

Constraints can be defined for each manipulated output: There is one set of hard absolute constraints, and one set of hard speed constraints. There is also one set of soft constraints with priorities that can be used to define the most desired operational range for manipulated outputs. The constraints can be defined either by external inputs or by the operator.

The operational mode of each constraint can be individually configured on-line, either to be defined by an external input or by the operator, or not to be accounted for at all.

Each soft constraint is given a priority that defines how important it is to satisfy a particular constraint in a situation where not all soft constraints can be satisfied. In such situations the optimization algorithm provides a control signal that minimizes a weighted sum of squared constraint violations. The weights in this optimization procedure depend on which of the nine priority levels that has been chosen.

Operator Macros

Macro functionality for coordinated setting of several parameters in the controller. The operator can define a list of actions for setting of e.g modes, set-points, and limits. The actions in the list are completed before the next sampling instant of the on-line controller. The list can be saved on disk for later retrieval.

Signal Linearization

The controller integrates functions for non-linear transformation on each signal. This is useful for e.g compensation of non-linear sensors and non-linear actuators.

Parameter Scheduling

The controller can be configured to use an external input signal for parameter scheduling. One of four pre-defined sets of controller parameters is then used in the algorithm of the controller. The set of controller parameters accounted for in the scheduling involves e.g a complete process model, equivalent error and move intervals, and limit definitions.

On-line Design Calculation

The controller can be reconfigured on-line, either when a new set of controller parameters is loaded or when the operational modes of signals or modes are changed.

PRBS Generator

The controller has functions for generation of excitation signals for process identification. Pseudo Random Binary Sequences can simultaneously be added to several manipulated outputs. This integrated function provides a user easily to do experiments for process identification.

Data Logging

The controller has data logging function that stores selected signals and variables in a log-file. This integrated function provides a user easily to do experiments for process identification.

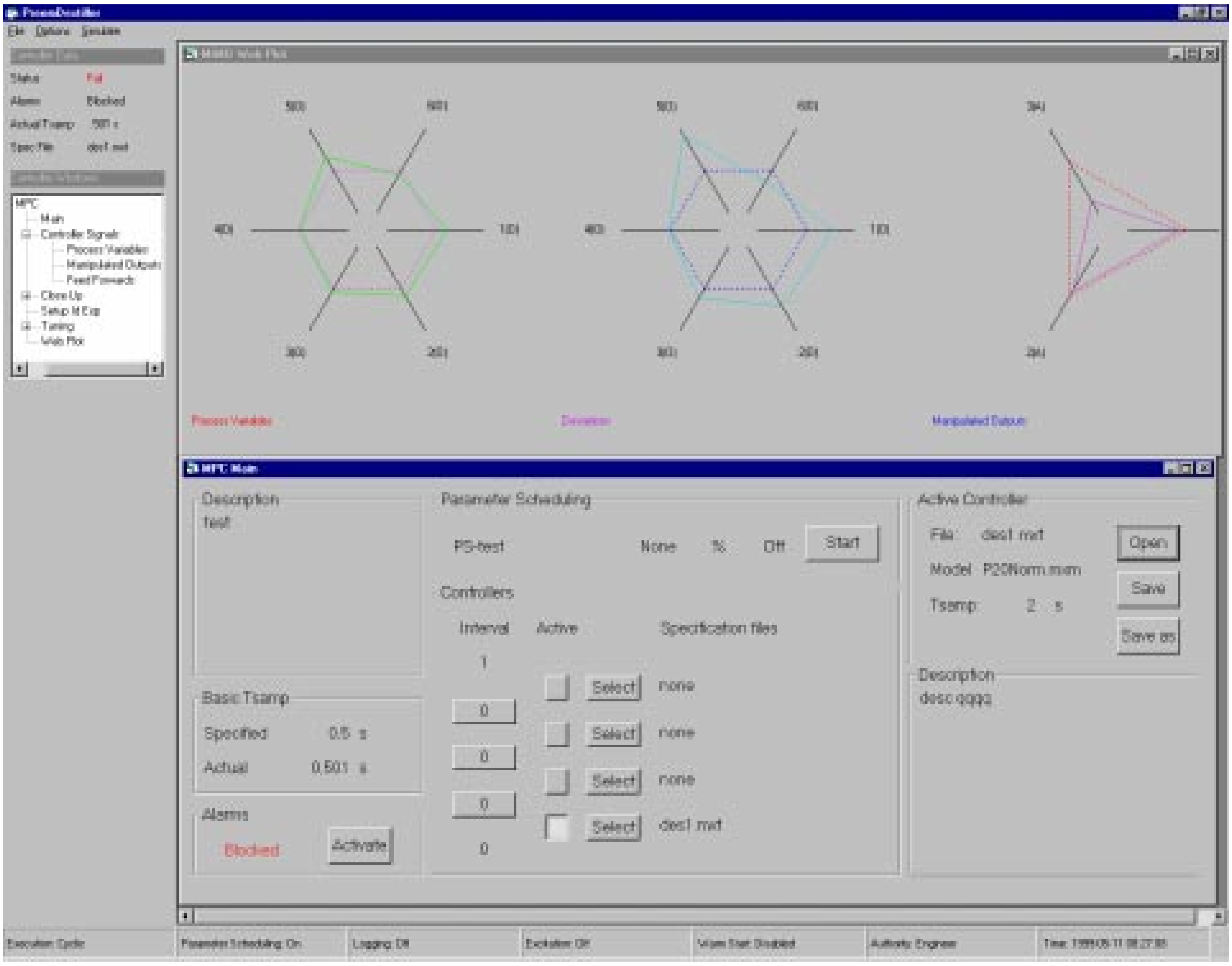
2.2.2 Graphical User Interfaces

Interface for start-up

The start-up interface is used to point out an application that will be controlled. The application has earlier been prepared by the off-line engineering tools. The start-up interface is also used to select the OPC-server that will be used.

Operator Interface

The operator interface presents actual data for the on-line controller and allows the operator to set parameters in the on-line controller.



Description
test

Parameter Scheduling
PS-test   None   %   Off

Controllers

Interval	Active	Specification files
1		
0	☐	none
0	☐	none
0	☐	none
0	☐	des1.mxt

Basic Tsamp
Specified   0.5 s
Actual   0.501 s

Alarms
Blocked

Active Controller
File: des1.mxt
Model   P20Norm.mxm
Tsamp   2 s

Description
desc qqqq

Figure 2-2. One of the operator interface displays.

2.3 Off-line Engineering Tools

The engineering tools are available in MATLAB. MATLAB is launched with the proper settings from the NT Start-Menu. The different engineering tools are then accessed using pull-down menus in the engineering tool main window that is available when MATLAB has been started. The following off-line engineering tools are available:

- Configuration Tool
- Pre-processing Tool
- Modeling Tool
- Tuning Tool.

These are described in the following sections. They all have advanced graphical user interfaces.

2.3.1 Configuration Tool

The configuration tool is used to create and modify the actual configuration of the control system. A configuration has to be defined before any of the other parts of the 3*d*MPC is used. The configuration is a definition of what signals in the OCS/DCS that are used as manipulated outputs, as process variables, and as feed-forward variables.

2.3.2 Pre-Processing Tool

The pre-processing tool provides functions for signal processing to generate data sequences suitable for the process identification procedure in the modeling tool. Data sequences are initially created from log-files that contain data that has been recorded by the on-line controller logging function.

The functionality available for signal processing is described below. Properties of the data sequences are shown graphically. The plots are updated after each action.

Linearization

Static non-linearities can be introduced for each individual signal to compensate for e.g non-linear sensors and non-linear actuators. These non-linear functions are preserved in controllers based on models built from the data sequences where the non-linear function originally was defined.

In the pre-processing tool the non-linear transformations of a data sequence are performed prior to the linear filtering.

Linear Filtering

A data sequence can be filtered by a low-pass, a high-pass, a band-pass, or by a band-stop filter. The filtering function also involve removal of mean-values and trends.

Data Editing

The data editing functionality allows removal of outliers in the data. A cut function removes a part of a data sequence, a split function splits a data sequence in two parts, and a merge function merges data sequences. There is also a function to re-sample a data sequence.

2.3.3 Modeling Tool

The modeling tool is used for building models of the process from data sequences. The models are intended for the controller tuning described in Section 2.3.4, Tuning Tool. There are modeling tools for process identification and for merging of models obtained from different process identifications. The tool also contains elaborate analysis functions to assist the user. The modeling tools are accessed using advanced graphical user interfaces.

Parametric Identification

The main part of the modeling tools are the parametric identification tool that is used for building state-space models from data sequences. The used two step state-of-the art algorithm for parametric identification combines the ease of use with the ability to use data sequences from both open and closed loop identification experiments.

Model Merging

There is a tool to merge models created at different occasions and covering different parts i.e. containing different sets of inputs and outputs of a process.

Given two models, a new model is constructed where each transfer function is taken from either of the two models. The user selects the used parts.

Definition of Simple Models

This tool allows a user to define models that are already known. For each component of a multivariable transfer function, the user can enter a gain, a time constant and a delay.

Model Evaluation

In addition to the model creation tools, there are also tools to examine the models in different respects. The following analysis functions are available:

- Simulated/Predicted output

  The measured outputs of a given evaluation data set are plotted together with either the simulated or the predicted model outputs.

- Prediction Error Variances

  The prediction error variances as a function of prediction length are plotted for given models and a given evaluation data set.

- Noise Spectrum

  The noise spectrum for each output of given models are plotted.

- Frequency Response

  Gain and phase curves for all transfer functions of given models are plotted.

- Transient Response

  Step or impulse responses for all transfer functions of given models are plotted.

- Poles/Zeros

  The poles and zeros for all transfer functions of given models are plotted.

2.3.4 Tuning Tool

The tuning tool is a graphical tool used for the controller tuning. The tool also contains elaborate analysis functions to assist the user.

MPC Tuning

This tool is used to create and modify data for the controller tuning.

MPC Tuning Analysis

The MPC tuning analysis tool is used to evaluate the behavior of the feedback control system. The following functions for analysis are available for all relevant open and closed loop transfer functions:

- Step responses
- Bode diagrams
- Nyquist diagrams
- Singular value frequency response plots.

Special emphasize is put upon transfer functions that are relevant for robustness considerations.

Chapter 3 Software Components

The product 3*d*MPC consist of on-line components for control and operator interaction, and off-line engineering components.

3.1 On-Line Components

The following on-line components are included in the 3*d*MPC product:

On-line Controller	The on-line model predictive controller functionality. It can operate in two different ways: • Real time control where data is exchanged with a DCS/OCS using an OPC server in the work station. • Simulations with an internal simulator.
Operator Interface	Graphical interface for modification and supervision of parameters and signals in the On-line Controller. Several Operator Interfaces may be connected to a specific On-line Controller. They may be located on any work station in the network. In case of off-line simulations an extra display is added for the control of the internal simulator

3.2 Off-Line Components

The off-line engineering components are available as a MATLAB toolbox. Although MATLAB is used as the computational engine, no MATLAB knowledge is required since all actions are done using the graphical interface.

3.3 Directory Structure

The 3*d*MPC product resides in a directory with the sub-directories according to Table 3-1.

Table 3-1. 3*dMPC* sub-directories.

Directory name	**Contents**
applications	Parent directory for the application root directories.
bin	Executable files for the on-line components.
etc	Miscellaneous files.
matlab	The MATLAB toolbox.

Chapter 4 Requirements and Compatibility

4.1 Hardware Requirements

In Table 4-1 you will find the minimum requirements for the 3*d*MPC.

Table 4-1. Minimum recommended hardware for 3dMPC

Component	Requirement
Personal computer (PC)	Pentium Pro II 350
Primary memory (RAM)	128 Mb
CD	yes
Color monitor	19” (1280 x 1024)

The selected personal computer must be designed to withstand climatic, mechanical and electrical effects in the intended environment.

4.2 Operating System Requirements

Windows NT version 4.0 with service pack 4 or later.
Windows 95/98 is not supported.

4.3 Software Requirements

MATLAB 5.2 for Windows, or later, is required for the engineering tools.

An OPC server is required for data transfer between the on-line controller and the underlying instrumentation system.

NOTE

The hardware requirements for the OPC server must also be considered.

4.4 Compatibility

The OPC clients supports the following interfaces:

- Data Access Standard 1.0a
- Data Access Standard 2.0.

Chapter 5 Environmental Immunities

The selected personal computer for the 3*d*MPC must be designed to withstand climatic, mechanical and electrical effects in the intended environment.

The general immunity of Advant OCS to environmental factors is summarized in *Product Guide Advant OCS with Master SW, Overview.*

The Advant OCS system has been designed to withstand the environments encountered in the power and process industries with respect to climatic, mechanical and electrical effects.

Products in the Advant OCS range, manufactured by ABB or others, are normally type tested or evaluated to ensure compliance with the environmental requirements specified below.

These requirements can be categorized as follows:

- Resistance to the external environment outside enclosures (including corrosive gases)
- Resistance to the environment inside enclosures
- Electro Magnetic Compatibility
- Electrical safety.

NOTE

The environmental specifications applicable to third party equipments, for example, PC:s and printers, may differ. Refer to the data sheets of the products concerned.

The degree to which the units withstand environmental influence depends on their components and their design. The system enclosures, the types of cable and how these are routed determine the environmental requirements of the complete system.

Chapter 6 Technical Data and Performance

6.1 Technical Data

The 3*d*MPC product can be configured to handle various numbers of signals. Table 6-1 shows the maximum number of each signal type that can be handled

Table 6-1. Maximal number of signals

Type of Signal	Maximum number
Manipulated outputs	16
Process variables	32
Feed-forward variables	16

6.2 Performance

The performance of 3*d*MPC depends on the chosen hardware. To give an indication of the memory consumption and the execution time a number of applications were tested on a 400 MHz Pentium II PC with 128 MB RAM. The results below are not claimed to be exact measures, but should rather be seen as indications of expected performance.

6.2.1 Memory Consumption

The memory consumption depend on the problem complexity. Figure 6-1 shows the estimated memory usage as function of the problem complexity. The solid line shows the estimated memory usage for continuous operation. The dashed line shows the estimated peak memory usage that occurs during an on-line redesign of the controller.

The complexity parameter n_1 is given by

$$n_1 = n_{out} \cdot m + n_{pv} \cdot p$$

where n_{out} is the number of manipulated outputs, m is the control horizon, n_{pv} is the number of process variables, and p is the prediction horizon.

6.2.2 Execution Time

The execution time per sample, for a certain hardware configuration, depend on the problem complexity. Figure 6-2 shows the estimated execution time per sample as function of the problem complexity for a PC with a 400 Mhz Pentium II processor

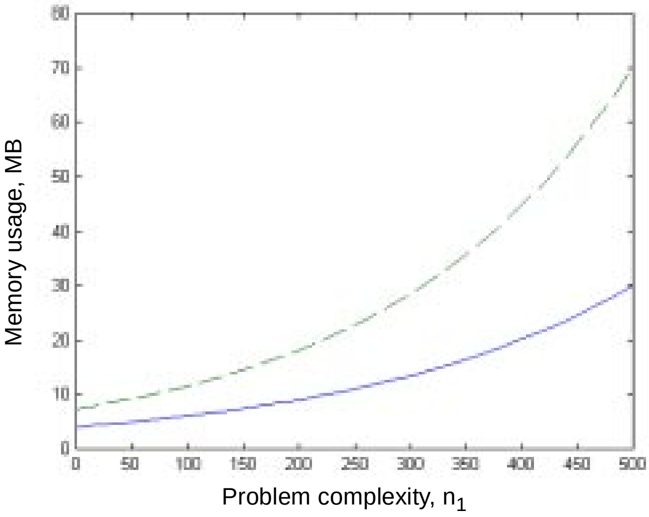


Figure 6-1. Estimated memory usage.

The complexity n_2 is given by

$$n_2 = n_{out} \cdot m + n_s + n_{db} \cdot p$$

and n_{out} is the number of manipulated outputs, m is the control horizon, n_s is the total number of soft constraints for process variables, deviations, and manipulated outputs, n_{db} is the number of dead-bands used for process variables, and p is the prediction horizon.

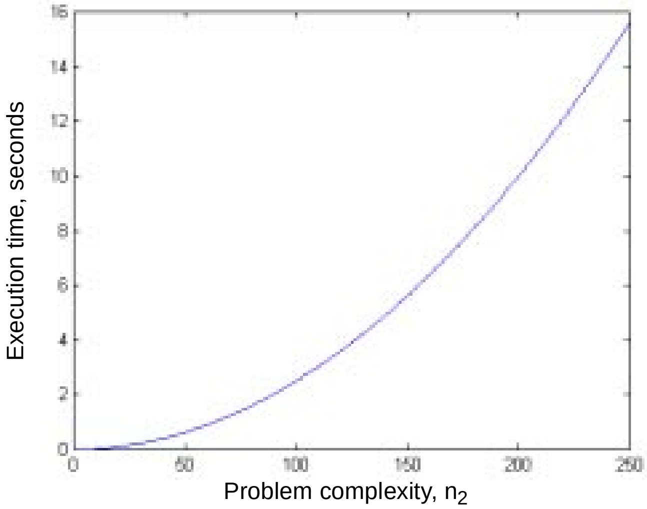


Figure 6-2. Estimated execution time.

Chapter 7 Ordering

For details, see the valid price list.

3*d*MPC Software Licenses

Table 7-1. 3dMPC Licence

Description
3dMPC On-Line Controller, License for ≤5 manipulated outputs
3dMPC On-Line Controller, License for ≤12 manipulated outputs
3dMPC On-Line Controller, License for >12 manipulated outputs
3dMPC Engineering Suite, One User license

Media

Table 7-2. 3dMPC Software Delivery Media

Description
3dMPC SW Distribution Media

Documentation

Table 7-3. Documentation on 3dMPC.

Description
3dMPC User’s Guide

Size Upgrade within Same Version

Table 7-4. 3dMPC size upgrade

Description
3dMPC On-Line Controller, License upgrade from ≤5 to ≤12 manipulated outputs
3dMPC On-Line Controller, License upgrade from ≤5 to >12 manipulated outputs
3dMPC On-Line Controller, License upgrade from ≤12 to >12 manipulated outputs

Version Upgrade

Table 7-5. 3dMPC version upgrade

Description
3dMPC On-Line Controller, Latest Version Upgrade. License for ≤5 manipulated outputs
3dMPC On-Line Controller, Latest Version Upgrade. License for ≤12 manipulated outputs
3dMPC On-Line Controller, Latest Version Upgrade. License for >12 manipulated outputs
3dMPC Engineering Suite, Latest version Upgrade. One User license

3BSE 018 935R101

January 2000